IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 July 2023, accepted 22 August 2023, date of publication 5 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3310470

RESEARCH ARTICLE

Deep Contextual Grid Triplet Network for Context-Aware Recommendation

SOFIA AFTAB, HERI RAMAMPIARO, HELGE LANGSETH, AND MASSIMILIANO RUOCCO
Department of Computer Science (IDI), Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway
Corresponding author: Sofia Aftab (sofia.aftab@ntnu.no)

This work was supported by the IDI Department of NTNU.

ABSTRACT Modeling contextual information is a vital part of developing effective recommender systems. Still, existing work on recommendation algorithms has generally put limited focus on the effective treatment of contextual information. Moreover, adding context to recommendation models is challenging since it increases the dimensionality and complexity of the model. Therefore, an efficient learning method is required to extract an association and inter-relationship between user/item features and contextual features for preference-driven modeling. The engineering of features through the exploration of adjacent correlations between the user/item and their context, and their further learning through a distance-based metric, is critical for effective personalization. Motivated by this, we introduce a context-aware recommendation strategy using a 'contextual grid triplet network.' This strategy uses a contextual grid topology to capture robust semantic representations of users, items, and contextual data. We present a learning methodology that merges a triplet network with a convolutional neural network. This fusion enables the exploration of associations both 'within' the contextual grid, such as between users or items, and 'between' different contextual grids, like between a user and items of input. Moreover, we present a variant of a hinge loss function using a triplet network for improved performance and fast convergence. In this work, we study how these aspects boost the quality of top-N recommendations. Furthermore, We show through extensive ablation-based experiments that the proposed method outperforms existing state-of-the-art techniques, demonstrating its robustness and feasibility.

INDEX TERMS Recommender systems, context-awareness, deep learning, triplet network, hinge loss.

I. INTRODUCTION

The success of recommender systems generally depends on their ability to capture and map user-item interactions into low-level features. In the quest for finding ways to learn the best features, deep metric learning methods have previously been applied [1]. In recommendation, distance metric methods can be trained to discriminate between user and item features. In this respect, distance metric methods using pair-wise learning have demonstrated good performance in solving top-N recommendation problems [2], [3]. On the other hand, point-wise methods are generally unable to handle the learnable parameters like distance metric-based features, which usually involve learning similarity or relationship between a user and items [1].

There is a significant amount of research on using context and user-item interaction for improved recommendations. Recommender systems, especially content-based recommender systems, generate more relevant and personalized recommendations by considering specific contextual information. Here, it is important to mention that the context used in our method differs from what is generally applied in the traditional content-based recommendation. In this work, we exploit ''user behavior'', which includes traditional user profile information, such as demographics, and information about the user's purchasing habits. On the other hand, most content-based recommender systems only consider the features/context of items (regardless of the user's preferences) and explore the similarity between them. Generally speaking, in content-based recommendations, only metadata/context of items are considered to generate recommendations, and the user's behavior is not considered. In this paper, we treat *context* as ''fully observable'' feature vectors known a priori,

The associate editor coordinating the review of this manuscript and approving it for publication was Fu Lee Wang.

consisting of a static vector of the user's profile information. This means that the availability of information at the time of a (purchasing) activity is essential for making correct predictions or generating relevant recommendations. Note that these contextual user features are different from context-independent user features – such as those resulting from user-item collaborative filtering since the latter are generated through interaction with items and ignore the additional relevant factors, including user profile and demographic information, thus yielding less powerful features and less predictive power. Moreover, using the user's profile information as a context can be considered an effective tool to mitigate the cold start problem of recommender systems.

Many of the proposed models in context-aware recommendation use factorization algorithms, e.g., [4], [5], and [6], or apply hand-crafted features, such as the effect of time on video recommendation [7]. However, research in recommendation systems has focused more on exploring specific kinds of contextual data than on applying user's contextual data [8]. For example, geographical and temporal data have been explored in depth [9], while other previous approaches based on deep neural network (DNN), e.g., [10] and [11], have often ignored context or largely relied on incorporating context as direct features. In summary, while applying contextual features (as direct features) using, for example, point-wise methods provide limited benefits [8], exploiting context with pair-wise methods is still not fully explored. Hence, these methods might miss important information that could be extracted from the available features to improve the recommendation quality.

Conventional methods often treat context as an afterthought or rely solely on item metadata. Moreover, they often overlook the intricate relationships between users, items, and context. For example, point-wise methods struggle to capture these relationships, while content-based systems rely solely on item features, neglecting user behavior. These limitations reflect the inability of traditional methodologies to model multiplicative effects and cross-features and there is a need to capture the local relationships within the data. For example, consider a localized region of the user's occupation (''teacher''), the user's age (''middle-aged''), and a movie's genre (''documentary''). There is a need for a feature detector that fires strongly when it sees this combination, effectively learning the ''cross'' between these three features. There should be an effective way to recognize that middle-aged teachers have a strong preference for documentaries, which is a specific multiplicative relationship between these three features. As a result, there exists a gap in effectively harnessing context-driven preferences, necessitating innovative methodologies like ours.

To treat contextual data effectively, especially when using pair-wise methods, we develop a new learning method, inspired by Siamese network [12], that enables a triplet network to learn the distance-based features. Our novel approach structures inputs as a grid, akin to an image, and utilizes Convolutional Neural Networks (CNNs) to uncover intricate relationships within user, item, and contextual features. The proposed architecture, *contextual grid triplet network*, accepts three inputs, namely user, u; a positive item for this user, p; and a negative item for this user, n; along with their context, c. Intuitively, p is more relevant to u than n. Using a convolutional neural network (CNN), the triplet network exploits the input information by making a grid topology of each input along with its context. This means that each contextual-grid input can be represented as (u, c), (p, c) and (n, c). We call this input structure a *contextual triplet*.

The intuitions behind integrating a triplet network and a CNN is two-fold: (1) to extract context-augmented features and (2) to learn and compute the similarity between users and items. Context-augmented features encode an association within the contextual grid, i.e., user-to-context or item-to-context, computed by learning the semantic structure *within* embeddings or low-level features. These low-level features are generated through CNN convolutions over a grid topology of inputs. As shown later, *contextual triplet* is an appropriate input structure for the CNN to exploit adjacent correlations. Further, the similarity between users and items is the association between contextual grids, i.e., user-to-item, computed by learning the distance metric *between* embeddings or high-level features. These high-level features are derived (using distance metric) from combinations of low-level features (extracted through convolutions) and involve a higher level of interpretation and understanding of the relationship between the context-augmented user and item features. These two kinds of associations are required to capture the fine-grained similarities between pairs of items [13], especially if the items are from the same category or are very close to each other in the embedding space.

To highlight the importance of different modules of the proposed method and to quantify the contributions to the performance improvement, we perform an ablation study [14], in addition to comparing against the state-of-the-art methods. An ablation study typically refers to systematically removing specific parts of the model or algorithm and then studying their effects on the performance. With the ablation variants of the *contextual triplet* and the *contextual grid triplet network*, we can show that the proposed modules outperform all other alternative variants by a significant margin.

We summarize our major contributions as follows:

- We introduce a new grid formulation of the input structure, called *contextual triplet*, for effective learning and representation of user-item characteristics along with contextual information.
- We propose an advanced variant of triplet network, *contextual grid triplet network*, for learning context-augmented features and with-in and between grid interactions simultaneously.
- We propose an improved version of the hinge loss function, i.e., the hinge distribution loss function, which boosts the convergence rate and improves the performance by exploiting the distribution of previous training batches.

- We perform an extensive comparative study, comparing point-wise vs. pair-wise and context-aware vs. non-context-aware learning methods, to evaluate the effectiveness of our proposed approach.
- We carry out an evaluation strategy based on an ablation study, allowing us to thoroughly and systematically verify the quality of recommendation generated using our method and its robustness.

The remainder of the paper is organized as follows. Section II gives an overview of existing methods and discusses their relation to our approach. Section III describes the motivation for the proposed method. Section IV describes the contextual grid triplet network approach in more detail. Section V outlines the experimental settings, including the dataset used, the evaluation metrics, and the details about the training. Section VI presents the results from our ablation-based experiments, which are further discussed in Section VII. Finally, Section VIII concludes the paper and outlines the future work.

II. RELATED WORK

Numerous studies have been conducted to develop effective recommender systems. In this section, we discuss those that we consider closely related to our work.

A. POINT-WISE METHODS

Many traditional recommendation algorithms employ point-wise learning, which aims at predicting the rating/score for each item. These include the collaborative filtering methods [15], as for example, matrix factorization [16], probabilistic matrix factorization [17], factorization machines [18] and nearest neighbours algorithms. Content learning methods [19], Deep learning methods [20], and some hybrid and integrated methods [21], [22], [23] (traditional methods combined with deep learning methods) also come under this category based upon the input data and loss/optimization functions. Point-wise methods mainly need an additional step (sorting) to generate a ranking of the predicted scores and are more related to a classification than a ranking problem.

B. PAIR-WISE METHODS

Pair-wise methods deal with ranking items, thus eliminating the need for sorting as with point-wise methods. An interesting aspect of pair-wise methods is that they capture the user's interest and preferences without generating quantitative ratings/scores. In comparison, point-wise methods ignore this personalized preference structure. Recently, pair-wise learning methods have been proposed to produce top-N recommendations by introducing and optimizing the preference structure between the original matrix and predicted matrix [2], [3], [24], [25]. Pair-wise learning methods treat data as pairs with which users can prefer positive items over negative ones. Different types of loss functions have been explored in these works, including the work by [2], which optimized the AUC score; the approach by [26], which explored the relationship between discounted cumulative gain and binary classification by mapping a ranking problem into the binary problem; and the method by [3], which exploited the hinge ranking loss to reduce the ranking risk in the reconstructed recommendation matrix. Similarly, [27] explored this ranking using a pair-wise metric learning approach.

C. CONTEXTUAL RECOMMENDATION

The terms contextual data and side information have been used interchangeably. There has been a tremendous amount of research on recommendations considering contextual data and side information. Likewise, the importance of modeling contextual information for recommendation is widely recognized. Nevertheless, particular types or aspects of contextual data have been explored in major recommendation domains, and many of these have been untouched or barely utilized [8]. A widely used type of contextual data is temporal dynamics [9]. For example, [7] explored the temporal dynamics effects in Netflix data and used these raw features directly in collaborative filtering. Similarly, the geographical context has been widely explored with matrix factorization [28] and tensor factorization [4]. Recently [29] has explored users' sequence of check-ins (venues) to model the users' short-term preferences Collaborative filtering methods have been generalized by factorization machines [30] and other contextual recommender systems [29], [31], [32]. While temporal and geographical dynamics have been largely explored, user profile dynamics have been a relatively less treated area within recommender systems.

The work presented in this paper extends existing work on context-based recommendation by incorporating the user's side information in the pairwise method, more specifically, a triplet network for improved recommendations. Here, we introduce an advanced variant of triplet network, called *contextual grid triplet network*, which improves the quality of recommendations.

III. MOTIVATION

Modern recommendation systems face the dual challenge of navigating complex high-dimensional user and item spaces while also taking into account relevant context to make accurate recommendations. Conventional approaches often rely on concatenating user and item features and contextual information, treating them as flat (1D) inputs. However, such methods may miss intricate relationships and high-order interactions between these different types of data. To tackle this, we propose a novel approach of structuring these inputs as a grid (2D), similar to an image and applying Convolutional Neural Networks (CNNs) to detect local patterns and relationships. We show that more complex representations like a grid lead to better performance but these models might take a bit longer training times. This motivates us to introduce a new variant of hinge loss for fast convergence. This new variant uses density distribution with a dynamic parameter α to adjust the margin rather than keeping it static as used in the original Hinge loss.

The grid-based approach has several advantages. It enables the explicit preservation of local interactions within the user, item, and contextual features, much like spatial relationships in an image. This structure allows the model to recognize and learn from these local relationships, facilitating a more comprehensive understanding of the interactions that drive user preferences. Using CNNs with this grid input further amplifies the model's ability to extract meaningful correlations. CNNs are inherently capable of capturing local dependencies and high-order interactions due to the hierarchical nature of their architecture. They can learn simple relationships by extracting low-level features(using lower layers) and use them to create high-level features (using higher layers) to understand more complex interactions. Moreover, the grid structure and CNNs together can naturally model common crosses and multiplicative relationships between features without needing explicit feature engineering. For instance, they could learn that the effect of a movie's genre (encoded in embedding) on a user's rating might depend on the user's age and occupation.

This capacity to model multiplicative effects and cross features offers a substantial advantage over standard feed-forward neural networks, which often struggle with learning these implicitly. By capturing high-order interactions and multiplicative relationships, our grid-based CNN approach provides nuanced modeling of user preferences. This has significant potential for improving the accuracy and relevance of recommendations, leading to more satisfied users and better engagement metrics. In the context of a rapidly evolving digital landscape, where personalized and context-aware recommendations are increasingly crucial, our method offers a promising avenue for advanced recommendation systems.

IV. CONTEXTUAL GRID TRIPLET NETWORK

In recommender systems, contextual features are typically used as concatenated inputs to neural networks. However, neural networks, especially the feedforward neural network, are inefficient in modeling the common crosses (multiplicative relations) and the association between inputs and contextual features [8]. To tackle this issue, we propose a network called *contextual grid triplet network* for generating recommendations.

Figure 1 shows the complete architecture of the contextual grid triplet networks. As shown in this figure, the architecture consists of two major modules: (1) embeddings learner which generates features $\Phi_E(.)$ and (2) regressor for rating predictions $\hat{y}$. We let U be the set of indexes over users, I denote the set of indexes over items and C be the set of indexes over context. Further, let u be a user, p and n two different items, and c a context, such that $u \in U$, $p \in I$, $n \in I$, and $c \in C$. Then, for any pair of contextual items given by $((p, c), (n, c))$, a contextual user, (u, c), has a preference denoted by $\succ_u$. Hence, $(p, c) \succ_u (n, c)$ implies that user (u, c) prefers item (p, c) over item (n, c). Together these three pairs constitute *contextual triplets* shown as grid formation in Figure1. More specifically, u, p and n can all be treated as vectors, as shown in Equation 1, where vectors represent different types of context for the same user. For simplicity, we will drop the explicit vector notation in the following and write, e.g., u, p and n instead of $\vec{u}$, $\vec{p}$ and $\vec{n}$. From the above preference relation, we can derive the desired output relation $y_{u,p,n} \in \{0, +1\}$, for each triplet $((u, c),(p, c),(n, c))$ as follows:

$$y_{((u,c),(p,c),(n,c))} = \begin{cases} 1 & \text{if } (p, c) \succ_u (n, c) \\ 0 & \text{otherwise.} \end{cases}$$

The rationale behind *contextual triplets* is to jointly explore the relationship between context vectors and user/item vectors. How the user/item embeddings evolve in relation to contextual information can explain the underlying association between context and user/items.

The contextual grid triplet network architecture shown in Figure 1 consists of three instances of the same CNN architecture (with shared parameters). The challenge is how to feed the *Contextual triplets* (u, c), (p, c) and (n, c) into the CNN to get meaningful representations of features. To learn context-augmented features, each entity of the triplet, i.e., u and c (where u is the user preference vector and c could be the age and occupation vector) is transformed into a grid topology. This grid topology can be defined as a matrix formed by stacking multiple contextual vectors at the bottom of the user or item vector for each input signal as used in Equations 2, 3 and 4 below. Here, $GNet(\cdot)$ is the function that applies two other intermediate functions, f^G and f^M, described later. $GNet(\cdot)$ generates intermediate embeddings Φ_E as represented in Equations 2, 3 and 4, where $P \in R^{M \times k}$, $Q \in R^{N \times k}$ and $X \in R^{D \times k}$ are learned latent factor matrices for users, items (p,n) and contexts respectively. As shown in the architecture in Figure 1, instead of generating raw embeddings, it generates embeddings as context-augmented user features and context-augmented positive and negative item features.

In Equations 2, 3 and 4, f^G is the function, for example, CNN, which accepts grid format input, consisting of user vector u *or* item vectors p and n along with context matrix c. F_r is a kernel in each convolutional layer, r, which performs convolution operation on grid format input. Here symbol "$*$" is a convolutional operator, b_r is a bias term and a is an activation function. The final set of output features vector from function f^G (using all kernels) is represented in the form of a set O, consisting of 1-D features $\{o_1, o_2 \ldots o_n\}$ which is fed into fully connected MLP network, represented as function f^M. It consumes input O from function f^G, where W is the weighting matrix, b is the bias term and a is an activation function from the respective layer, L.

Intermediate embeddings after applying f^M are fed into the triplet loss function, $Tri(\cdot)$, for calculating the distance from (u, c) to (p, c) and from (u, c) to (n, c) and give the final similarity score. This similarity score is used to generate the item preferences for the user, and these preferences are

encoded in the form of final learned embeddings. The L_2 distances (detail is in Section V-B) are represented in the triplet loss function, *Tri*, as show in in Equation 7.

$$c = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_t \end{bmatrix} \tag{1}$$

$$GNet(u, c) = f^M\left(f^G\left(\begin{bmatrix} u \\ c \end{bmatrix}\right)\right) = (\Phi_E(P), \Phi_E(X)) \tag{2}$$

$$GNet(p, c) = f^M\left(f^G\left(\begin{bmatrix} p \\ c \end{bmatrix}\right)\right) = (\Phi_E(Q), \Phi_E(X)) \tag{3}$$

$$GNet(n, c) = f^M\left(f^G\left(\begin{bmatrix} n \\ c \end{bmatrix}\right)\right) = (\Phi_E(Q), \Phi_E(X)) \tag{4}$$

$$\begin{aligned} f^G &= a\left(\left(\begin{bmatrix} u \\ c \end{bmatrix} \vee \begin{bmatrix} p \\ c \end{bmatrix} \vee \begin{bmatrix} n \\ c \end{bmatrix}\right)F_r + b_r\right) \\ &= \{o_1, o_2, \dots o_n\} = O \end{aligned} \tag{5}$$

$$f^M = a_L\left(W_L^T\left(a_{L-1}\left(\dots a_2\left(W_2^T * O + b_2\right)\dots\right)\right) + b + L\right) \tag{6}$$

$$Tri(GNet) = \begin{bmatrix} L_2(GNet(u, c)) - L_2(GNet(p, c)) \\ L_2(GNet(u, c)) - L_2(GNet(n, c)) \end{bmatrix} \tag{7}$$

As shown in Figure 1, embedding learning is a two-step process. The first step is the extraction of context-augmented features for user-context and item-context grid, i.e., Within-grid relationships using Equations 2, 3 and 4, where *CNN filters* convolve over the matrix to learn a fine-grained similarity, needed for learning features from the same category [13]. To illustrate, for item ranking, a user always prefers positive items over negative items and more specifically, more positive items over less positive items. Therefore, we usually want to rank items with a rating 'of 5' higher than those with a rating 'of 4', although both are positive. The intuition behind context-augmented features through contextual triplet is to explore how context affects the user's interest in items and how this context correlates with the items. An alternative approach is to explore how much a user can prefer a specific item with respect to his context and up to what extent an item can be considered preferred according to the context. For example, the model might learn that a particular user's context, i.e., occupation (let's say ''student'') in the user context often aligns with a preference for certain genres or types of movies (like action or sci-fi) in the user's embedding. On the item side, it might learn that movies with certain features (such as action movies) often align with high ratings among certain user occupations (like students).

The second step is to explore the between-grid relationships, i.e., when comparing the user-context and item-context grid, the model might recognize that users of a certain age or with a certain occupation tend to rate certain types of movies more highly. For example, it might be learned that users identified as ''students'' tend to give higher ratings to action movies. Furthermore, *triplet loss* in Equation 7, like any distance-based loss, tries to ensure that semantically similar data points are embedded close together. This means that a user is placed more closely with the positive items than the less positive items and eventually further away from the negative items in the embedding space. This way of placing users and items in the embedding space through distance metrics is a suitable way of representing the preference structure of items for a user. Item similarity relationship is characterized by relative similarity ordering in triplets [13].

The idea is first to extract the low-level context-augmented features to learn the powerful representation of triplets and then learn the high-level item-to-user similarity on top of these features. For example, in the user-context grid, a low-level feature might be a specific sequence of values (or range of values) that often appears together and signifies a particular user behavior. In the age vector, a low-level feature might be a certain pattern of ages that indicates a specific user group. On the other hand, the high-level feature could recognize a pattern where a certain sequence in the user vector (signifying a particular user behavior) often appears with a certain pattern in the age vector (indicating a specific user group). This could indicate that users from a specific group often exhibit certain behaviors. Similarly, in the occupation vector, another high-level feature might be a certain pattern that correlates with another pattern in the user vector, indicating that users with specific occupations tend to have specific behaviors. We also show these patterns in our proposed solution using t-SNE visualization in SectionVII.

Our proposed approach to capturing the preference structure is different from the traditional settings since the similarity between a user and items is computed over context-augmented (user-to-context/item-to-context) representations instead of using raw or concatenated features. As we show below, the learned embedding representations, $\Phi_E(\cdot)$, from the embedding learner are fed into a scoring function, f^S for final rating prediction as shown in Equation 8.

$$\hat{y}(u, i, c) = f^S(\Phi_E(.)) \tag{8}$$

The scoring function only requires the user and item i, along with the context, irrespective of positive and negative inputs, for rating predictions. This is why we use shared embeddings for positive and negative items. Hence, y is a scoring function of user, item and context resulting in a predicted score. We explain this scoring function in detail in Section V.

V. EXPERIMENTAL SETTINGS

We conducted a number of experiments aimed at evaluating how the learning of user and item representations, as well as the preference structure of users for items can be efficiently modeled with grid topology. Broadly, we performed three and more specifically, 11 different experimental settings to examine the robustness of the proposed model. Broader categories include (1) a point-wise approach, (2) a pair-wise approach, and (3) the proposed approach. For specific categories, in addition to state-of-the-art methods, we employed an ablation study with respect to contextual vs. non-contextual data and grid vs. non-grid topology treatment. Ablation studies are generally suitable to systematically investigate knowledge

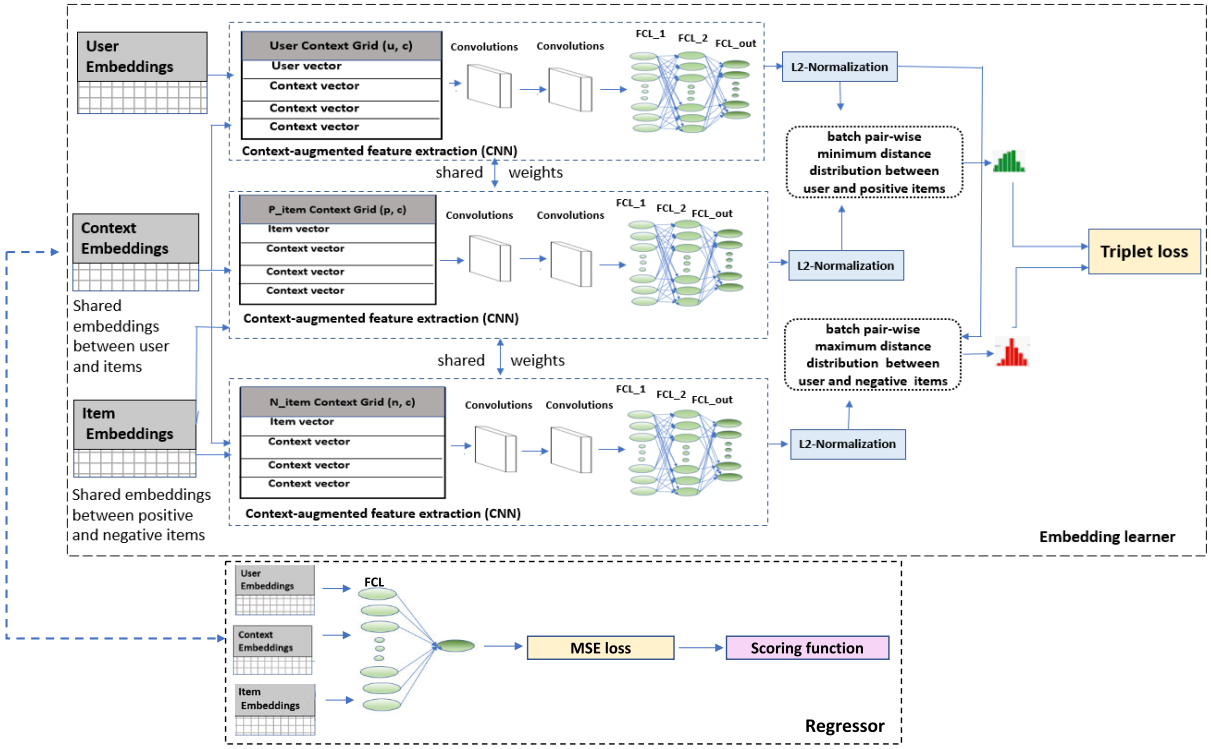


FIGURE 1. The architecture of the contextual grid triplet network: The embedding learner is learning the features by using grid input through triplet loss (using proposed hinge distribution). The Regressor uses the learned features and uses MLP for prediction.

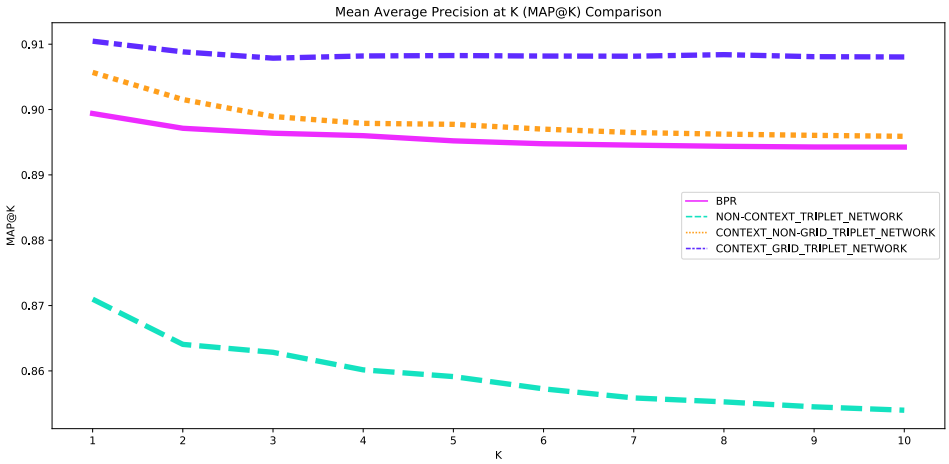


FIGURE 2. MAP@k for the pair-wise models on *ml-1m* data.

representations in the artificial neural network due to their ability to disclose specific parts of the representations that contribute to the effectiveness of the network [14].

A. DATASET AND EVALUATION METRICS

In our experiments, we used three data sets, two of which are different versions of MovieLens data sets. First is *MovieLens 1M (ml-1m)* and second is *MovieLens 25M (ml-25m)* which is the latest and more stable version (released in 12/2019) with ratings 1 to 5. The third one is *RentTheRunway* (RTR) [33] data set, the largest rental platform for women's clothing. It is relatively difficult to find rating data for recommendations with user profile information. This is why we use two versions of *MovieLens* and *RentTheRunway* data sets, available with user profile data. The dataset preparation method is different for the two types of learning methods since a point-wise method accepts individual inputs, while a pair-wise learning method accepts inputs in the form of pairs or triplets. For point-wise learning,

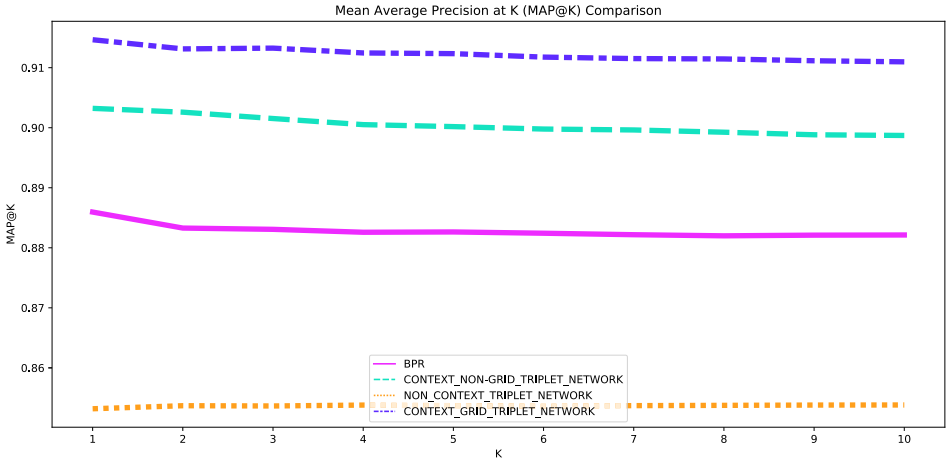


FIGURE 3. MAP@k for the pair-wise models on *ml-25m* data.

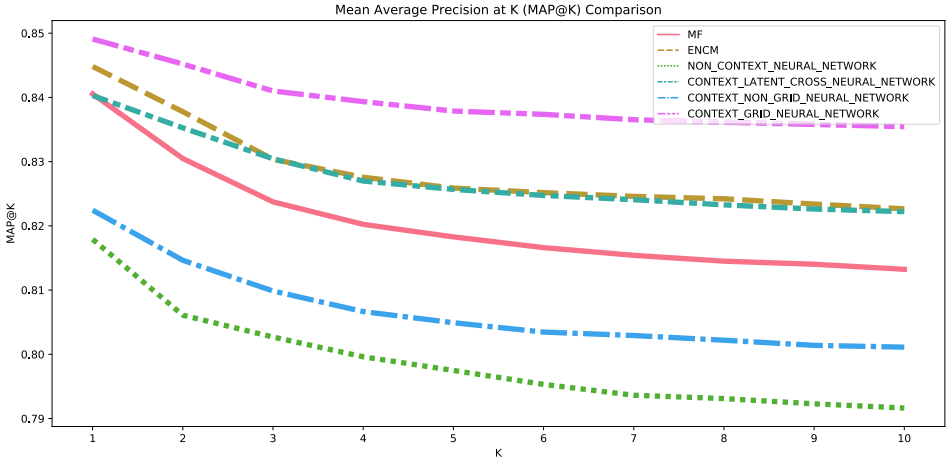


FIGURE 4. MAP@k for the point-wise models on *ml-1m* data.

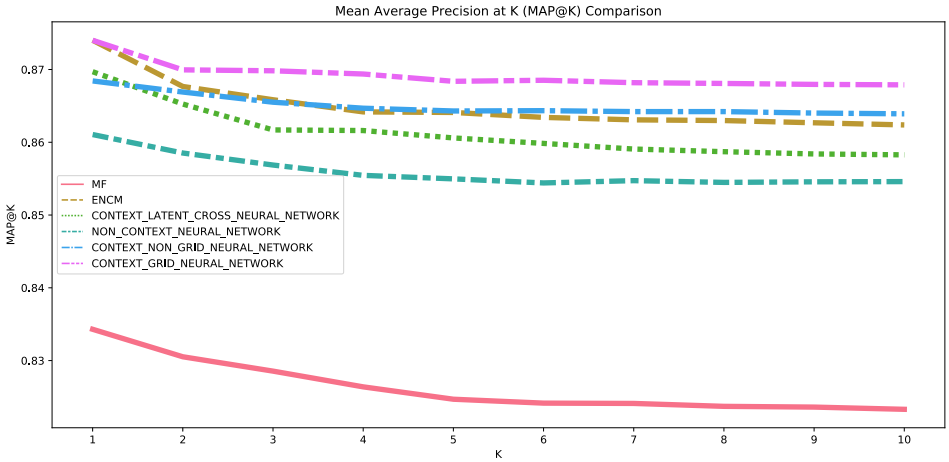


FIGURE 5. MAP@k for the point-wise models on *ml-25m* data.

we split the dataset into 70 % training set and 30% test set. For pair-wise learning, we first create the triplets, in which each user is coupled with positive items (extreme positive with ratings $\geq$ 5) and randomly chosen negative items (extreme negative with ratings ≤ 2). This method of creating triplets is specifically useful for learning discriminating features.

We used optimization methods like *hard triplet mining* [34] and L_2 normalization (see Section IV) to tackle some inherent problems/challenges like triplet selection and convergence

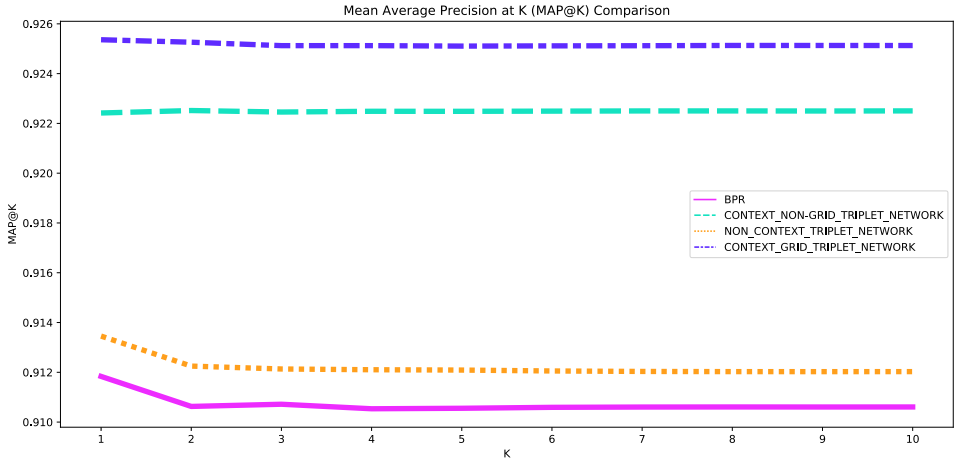


FIGURE 6. MAP@k for the pair-wise models on *RTR* data.

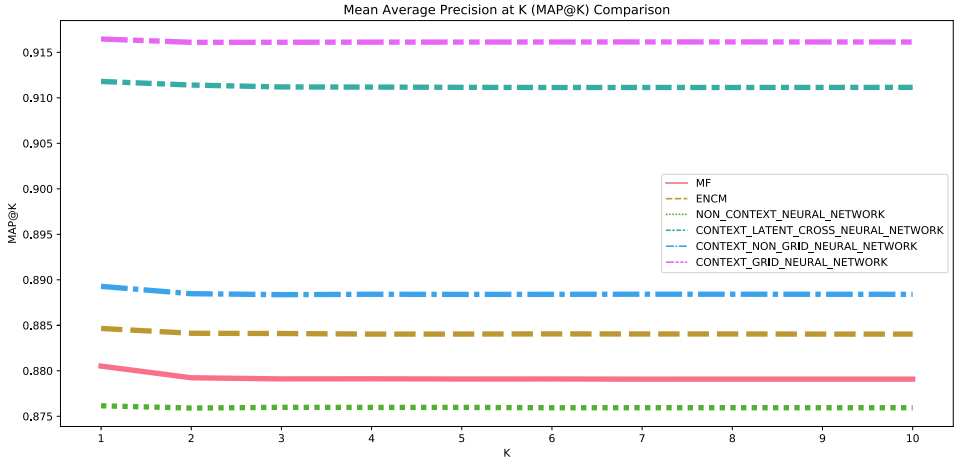


FIGURE 7. MAP@k for the point-wise models on *RTR* data.

TABLE 1. performance comparison against state-of the-art-methods.

Model Name	Type	Context	Grid	MAP(ml-1m)	MAP(ml-25m)	MAP(RTR)
MF	Point-Wise	No	No	0.8405	0.8342	0.8805
ENCM	Point-Wise	Yes	No	0.8448	0.8740	0.8846
Context-Latent-Cross-Neural-Network	Point-Wise	Yes	No	0.8402	0.8696	0.9118
BPR	Pair-Wise	No	No	0.8993	0.8859	0.9119
Non-Context-Triplet-Network	Pair-Wise	No	No	0.8709	0.8531	0.9134
Context-Grid-Triplet-Network	Pair-Wise	Yes	Yes	**0.9104**	**0.9146**	**0.9253**

time [34]. We applied a *hard negative* strategy for triplet generation, which was done by selecting the hard negative triplets among the batch of items, as proposed in [34]. We used all negative triplets where the distance from (u, c) to (n, c) is less than 0.5. After generating the triplets, the dataset was split into 70% training set and 30% test set. For fair evaluation, the triplets included in the test set were excluded from the training set.

We applied a standard quantitative metric, MAP@k [35], for the performance evaluation since we assumed binary relevance for the items, i.e., an item is either of interest or not. Moreover, using MAP to evaluate a recommendation engine implies that we can treat the recommendation task like a ranking task that needs a lot of "correct" or relevant recommendations earlier on in the list.

B. TRAINING DETAILS

The training involves two models, embedding learner and regressor. For the embedding learner, we use a convolutional network consisting of three convolutional and 1×1 average-pooling layers. The network configuration (ordered from input to output) consists of filter sizes {1, 2, 2} and feature map dimensions {150, 500, 500, 256}, where the 150 vector is the embedding dimension (input size) and the 256 vector is the embedded representation of the network followed by

TABLE 2. Performance comparison among ablation variants.

Model Name	Type	Context	Grid	MAP(ml-1m)	MAP(ml-25m)	MAP(RTR)
Non-Context-Neural-Network	Point-Wise	No	No	0.8179	0.8610	0.8761
Context-Non-Grid-Neural-Network	Point-Wise	Yes	No	0.8224	0.8684	0.8892
Context-Grid-Neural-Network	Point-Wise	Yes	Yes	0.8490	0.8740	**0.9175**
Context-Non-Grid-Triplet-Network	Pair-Wise	Yes	No	0.9056	0.9032	0.9222
Context-Grid-Triplet-Network	Pair-Wise	Yes	Yes	**0.9104**	**0.9146**	**0.9253**

two fully-connected hidden layers, each with 500 nodes and a Relu activation function. The final output layer is with 256 nodes and this 256 vector is the final representation of embeddings coming from each instance (u, c), (p, c), (n, c) of the triplet network (see Figure 1).

We apply an L_2 normalization to the embeddings before feeding them into the triplet loss function. The advantage of applying normalization like this can be compared to the advantage of cosine similarity to Euclidean distance. The squared Euclidean distance between normalized vectors is proportional to their cosine similarity, so the value of squared Euclidean distance is guaranteed to be within the range [0, 4] (cosine similarity value). The training is done in batches with a batch size of 64 and runs over ten epochs. The embedding layer dropout is fixed at 0.05, whereas the layer dropout is in order {0.5, 0.5, 0.25} from the hidden to the output layer. The learning rate is 0.005, and the momentum is 0.9. The model is trained using back-propagation with ADAM [36]. During each training pass, the embeddings are evolved and improved by using triplet hinge distribution loss at the end of the network. The existing triplet hinge loss [27] can be written as in the Equation below.

$$L_{tri}(\tau) = max\big(\big(||(u) - (p)||^2 - ||(u)) - (n)||^2 + \alpha\big), 0\big) \quad (9)$$

Finding an appropriate fixed margin α with hinge loss is hard. Therefore, we propose a new variant of the existing triplet hinge loss by introducing a data-driven margin. This margin is updated/adjusted using distance distributions for each batch rather than a constant margin as in triplet hinge loss, for fast convergence. This fast convergence is important for real-time recommendations. Hence, the proposed loss for each triplet τ is,

$$\begin{aligned} L_{tri}(\tau) = max\big(\big(&||L_2(GNet(u, c)) - L_2(GNet(p, c))||^2 \\ &- ||L_2(GNet(u, c)) - L_2(GNet(n, c))||^2 + \rho\big), 0\big) \end{aligned} \quad (10)$$

Here, ρ is a violation margin that requires the distance of negative pairs to be larger than the distance of positive pairs for each batch. Unlike α, a constant in triplet hinge loss, ρ is a placeholder calculated from positive and negative distance distributions using Equation 7 mentioned above. Since the triplet loss is a monotonically decreasing (distance) function and for each iteration, it causes a decrease in positive distance and an increase in negative distance from the user. We can use this information to determine the appropriate margin for the next batch. Therefore, ρ is updated for each batch, providing a data-driven margin for fast convergence. We provide further details in Section VII-C. For all the triplets in the training set, S, the final objective functionf^t to optimize is then given by

$$f^t = \frac{1}{2}\sum_{\tau \in S} L_{tri}(\tau)$$

Finally, after training the embedding learner, we extracted learned embedding matrices for the user as $\Phi_E(P)$, for items as $\Phi_E(Q)$ and for context as $\Phi_E(X)$, and train a simple 1-layer network model by formulating an input (ψ) as shown in equation 11. Here u, i and c are user, item and context vectors, used in scoring function f^S for final prediction $\hat{y}$, as shown in Equation 12. Here ϕ_{MLP} and ϕ_{out} are mapping functions from input to output. Since the scoring function, f^S is defined as a 1-layer neural network, it can be formulated as in Equation 13.

$$\psi = \big(\Phi_E(P^T)u^U, \Phi_E(Q^T)i^I, \Phi_E(X^T)c^C\big) \quad (11)$$

$$\hat{y}(u, i, c) = f^S\big(\psi \mid \Phi_E(P), \Phi_E(Q), \Phi_E(X)\big) \quad (12)$$

$$f^S(\psi) = \Phi_{out}\big(\Phi_{MLP}\big(\psi\big)\big) \quad (13)$$

VI. EXPERIMENTAL RESULTS

A. BASELINES

We compared the proposed method with two main categories of baseline learning methods, consisting of point-wise and pair-wise methods. Hence, within each of the categories, we conducted two types of experiments. The first type is defined as the most relevant state-of-the-art methods, while the other type investigates the effects of different ablations using deep neural networks on the proposed method. Recall that the main focus of this paper is to find an effective way to treat contextual information for improved recommendation. To achieve this, we implemented the method by enhancing the triplet network and by introducing a grid topology. Taking these ideas further, we explored and performed experiments using an ablation study. In this section, we present and discuss the empirical results of our experiments.

B. POINT-WISE METHODS

In point-wise approaches, embedding learning and rating prediction are done in the same training loop (single model). Matrix factorization (MF)-based methods, latent cross (LC) and explicit neural context model (ENCM) are the most relevant state-of-the-art methods within point-wise approaches, which makes them a natural choice for our baseline. Further,

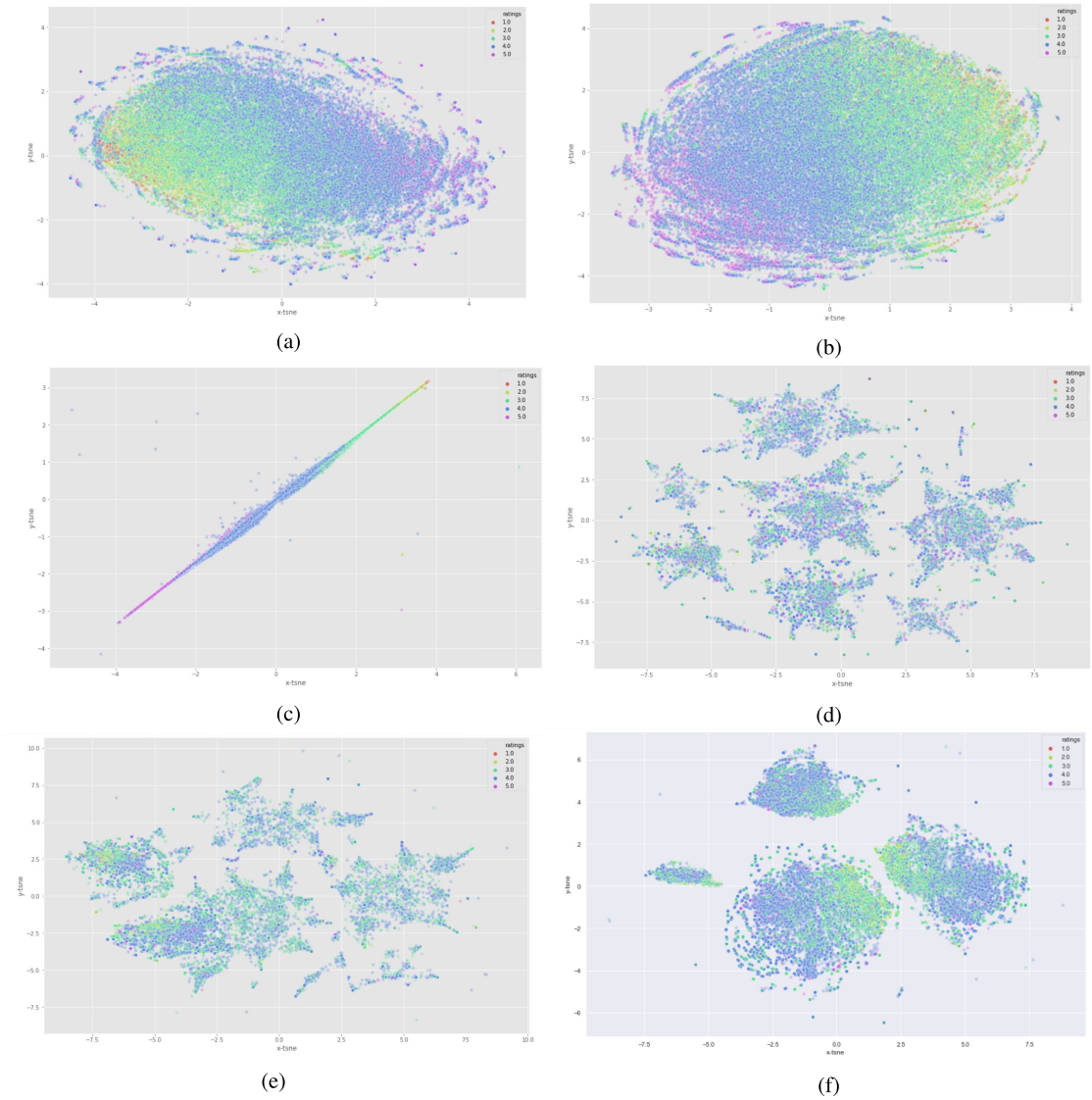


FIGURE 8. Items are clustered by rating using t-SNE components of embeddings: The top two figures are from applying the point-wise methods, with (a) using DNN, without context, and (b) using DNN with context. (c) is point-wise using CNN without context and (d) is point-wise using CNN with contextual grid. The bottom left (e) is the pair-wise method, the most relevant ablation variant to the proposed method, using CNN with non-grid context and finally, the bottom right (f) is the proposed method showing multiple patterns/clusters of preferences.

for our ablation study, we implemented several ablation variants based on our proposed method, incorporating contextual information and applying grid topology with CNN.

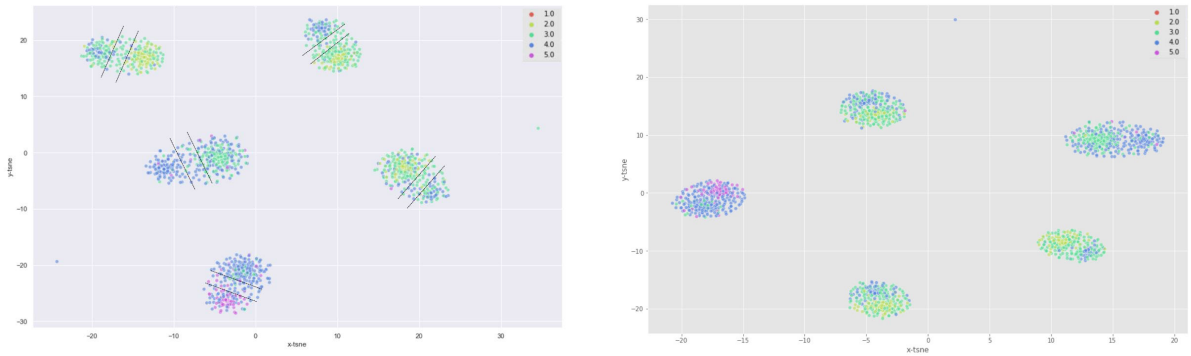
1) POINT-WISE STATE-OF-THE-ART MODELS

The point-wise-based baseline methods are outlined in the following:

- **Matrix factorization (MF):** Matrix factorization methods [37] learn low-rank decomposition of user-item interaction matrix by minimizing the square loss function. We specifically used this technique as a baseline to compute the preference degree by the product of user features and item features.
- **Context-Latent-Cross-Neural-Network (LC):** Latent Cross methods [8] treat contextual data differently. In the base paper, they use dynamic context in each time step of the LSTM unit and take the dot product of hidden vector and context vector; hence the name is Latent Cross. We use this method with CNN and point-wise loss instead of LSTM since we are not dealing with sequence-to-sequence learning.
- **Explicit Neural Context Model (ENCM):** This method is an extension of neural matrix factorization *(NeuMF)* models [38], [39]. This method feeds the explicit context vector to the standard NeuFM model, which is a generalization of MF to non-linear settings.

2) POINT-WISE ABLATION BASED MODELS

The point-wise-based ablation variants of models are outlined in the following:



(a) (b)

FIGURE 9. Items clustered by rating for top 5 users using t-SNE components of embeddings. Each cluster represents an individual user's rated items and it shows that the preferred items are separated by a considerable margin from the rest of the items in the proposed method (a) , as compared to the most relevant ablation variant (b).

- **Non-Context-Neural-Network:** We chose a CNN architecture as a baseline for rating prediction but it is different from our proposed approach in the input structure. In this method, we did not use contextual variables; thus using grid formation is not possible. Concatenated inputs (user, item) are fed into the CNN network and then fed into the linear layer, after flattening for final rating prediction.
- **Context-Non-Grid-Neural-Network:** In this method, we used concatenated features with CNN. User, item and contextual features are presented to the network.
- **Context-Grid-Neural-Network:** In this method, we have stacked the contextual vector with the user vector and item vector to make grid topology and this grid is presented to the network as an input signal. This method differs from our proposed idea since it uses a point-wise loss function for learning.

C. PAIR-WISE METHODS

In pair-wise approaches, two separate models were used: one for learning embeddings and the second to use these embeddings to predict the final ratings. We used embedding learning and a simple one-layer feed-forward neural network for predicting the ratings. BPR and a variant of the triplet network were selected as the most relevant state-of-the-art method within pair-wise approaches. This category employs contextual information and grid topology with pair-wise loss as ablation variants with CNN.

1) PAIR-WISE STATE-OF-THE-ART MODELS

The pair-wise based state-of-the-art methods are outlined in the following:

- **Bayesian personalized ranking (BPR):** Bayesian personalized ranking (BPR) [24] is a *loss* function and optimization method (a variant of stochastic gradient descent), which accepts inputs in the form of triplets, i.e., u, p, and n. It is the most relevant state-of-the-art method that can be compared to our approach. It takes the dot product of user features with all available items(positive and negative) features, making it the right choice for personalized ranking.
- **Non-Context-Triplet-Network:** We chose a CNN architecture, forming a triplet network with three instances. The triplets, u, p and n, are fed into instances of CNN to learn the embeddings. In the original paper [1], the authors used this triplet network for image instances. Here, however, we adapted it for user-item instances. Triplet loss is applied at the end of the network for learning the embeddings. We used this baseline to establish the effectiveness of contextual grid topology compared to a non-contextual CNN.

2) PAIR-WISE ABLATION BASED MODELS

The pair-wise-based ablation method is outlined in the following:

- **Context-Non-Grid-Triplet-Network:** In this method, we introduced contextual information to the triplet network as direct features. As mentioned earlier, we have extended the triplet network and enabled it to consume contextual information. The reason for using this architecture as a baseline is to establish and prove the effectiveness of contextual grid topology compared to contextual non-grid topology.

D. RESULTS

In this section, we present the results from our experimental evaluation, applying MAP@k.

The results using MAP@k for the pair-wise methods are reported in Figure 2, Figure 3 and Figure 6. Firstly, our results show that overall, the pair-wise methods perform better than the point-wise methods. Importantly, we can observe that our proposed approach *Context-Grid-Triplet-Network*, outperforms all other approaches. Secondly, focusing on the pair-wise methods (see Figure 2, 3 and 6), *Non-Context-Triplet-Network* has the lowest performance. Interestingly this method does not incorporate any contextual information. *BPR* performs better than *Non-Context-Triplet-Network* (relatively close to it in RTR data) but performs worse than

TABLE 3. Accuracy and AUC comparison in hing distribution Loss with the last margin for each epoch margin. It converges quickly even after the first epoch. Best results are in bold.

Epoch	margin (ρ)	accuracy	AUC
0	0.9648	0.7945	88
1	0.9749	0.8273	89
2	0.9751	0.8351	89
3	0.9757	0.8354	90
4	**0.9764**	**0.8434**	**90**

TABLE 4. Accuracy and AUC comparison in hing loss with fixed margin. It takes a longer time to converge. Best results are in bold.

Epoch	margin (α)	accuracy	AUC
0	0.5	0.5483	55
1	0.5	0.7804	86
2	0.5	0.8086	89
3	0.5	0.8335	89
4	**0.5**	**0.8364**	**89**

other methods. We note that these two methods do not incorporate context or grid topology. However, an ablation variant *Context-Non-Grid-Triplet-Network* performs better than all other non-context models but the proposed method outperforms it with grid topology. This means that applying contextual information in an effective manner contributes to improving the results, but not incorporating grid topology seems to make the method perform worse than the method that uses grid topology, i.e., *Context-Grid-Triplet-Network*. Overall, the proposed method *Context-Grid-Triplet-Network* has the best results, showing the impacts of incorporating both contexts and grid topology. The results for the point-wise methods are reported in Figure 4, Figure 5 and Figure 7. It can be observed, none of the point-wise methods perform better than the pair-wise counterparts, but it is worth noting that within the point-wise analysis, *Context-Grid-Neural-Network* is better than *Context-Non-Grid-Neural-Network* because of the usage of grid topology and *Context-Non-Grid-Network* is better than *Non-Context-Neural-Network* due to incorporation of contextual information (We also show the effectiveness of grid topology through embedding visualization in Section VII). However, recently proposed NeuMF-based extension, ENCM and latent cross neural network (LC), both incorporate context and show relatively better performance than the other baseline methods and ablation variants. Nevertheless the results from applying the proposed approach, i.e., *Context-Grid-Triplet-Network*, is still overall the best, hence showing its effectiveness within both pair-wise and point-wise methods.

E. PERFORMANCE COMPARISON AND CONTRIBUTION OF PROPOSED METHOD

In this section, we compare our proposed method against the state-of-the-art methods and provide the results of the ablation variants with respect to different components involved in architecture. With the ablation study, we show and quantify how different components, i.e., model type, category, context and grid topology, contribute towards the modeling performance, i.e., how the performance varies by adding or removing those components. The results of the comparison against the state-of-the-art methods are provided in Table 1, while Table 2 presents the results from our ablation study. We summarize our comparison and proposed method contributions as follows:

- Models applying context perform better than non-context models, and pair-wise contextual models show better performance than point-wise contextual models.
- Models with context-grid topology are generally better than those with context-non-grid topology, with both pair-wise and point-wise methods, but the improvements are larger with pair-wise methods, as compared to point-wise methods. This behavior is also depicted through embedding visualization in Figure 8 and Figure 9.
- All state-of-the-art methods have varied context usage, and our proposed ablation variants exhibit similar behavior as mentioned in the above two points.
- In state-of-the art-methods in Table 1, we show that models using context in an efficient way, for example, LC, ENCM and grid topology enable the model to learn better features than those using context as direct features. This is due to the fact that these methods explore similarity or multiplicative relationships between user/item features and contextual features. However, adding context as raw features does not help to extract the above-mentioned relationship. Moreover, grid topology outperforms the other state-of-the-art methods since it extracts correlation-based features using user/item and context which is an appropriate method to generate preferences over items. For example, a user's context may have an influence on the selection of items, and therefore, users who share similar contexts may prefer similar items.
- Table 1 and Table 2 show that models with context and grid topology in both point-wise and pair-wise methods outperform the state-of-the-art methods and ablation variants. This demonstrates the validity and effectiveness of the proposed method.

VII. DISCUSSION AND ANALYSIS

In this section, we explore a number of research questions and analysis scenarios considered by this work.

A. DATA PREPARATION

The main focus of this paper is contextual treatment of data, and in relation to that we have explored various methods and scenarios to investigate the similarity and multiplicative relationships between contextual features and model performance. Starting from data preparation to model evaluation, various techniques have been used to improve every step of modeling. For example, since contextual data, age and occupation are categorical variables, we applied binning transformation after mapping them to numerical values. In other words, each bin represents a certain group of people.

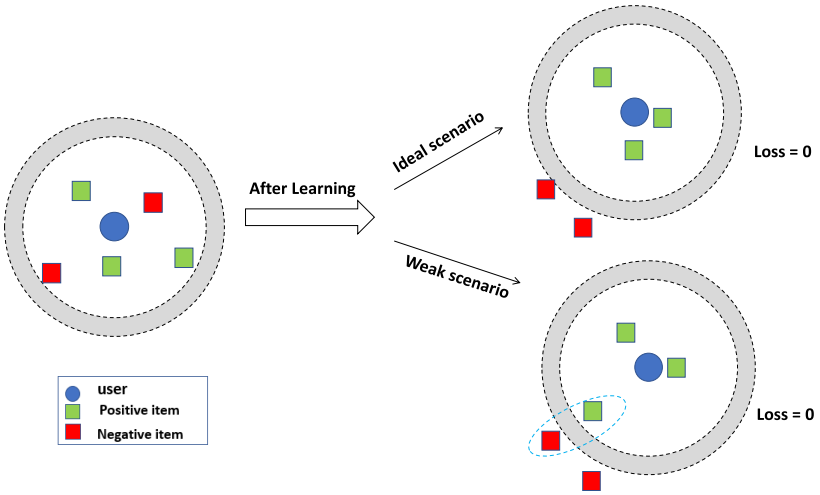


FIGURE 10. Distance learning using hinge loss function. The distance of a positive item from a user, including the margin is slightly smaller than from the user to a negative item, and thus hinge distribution loss assigns a zero loss to this pair.

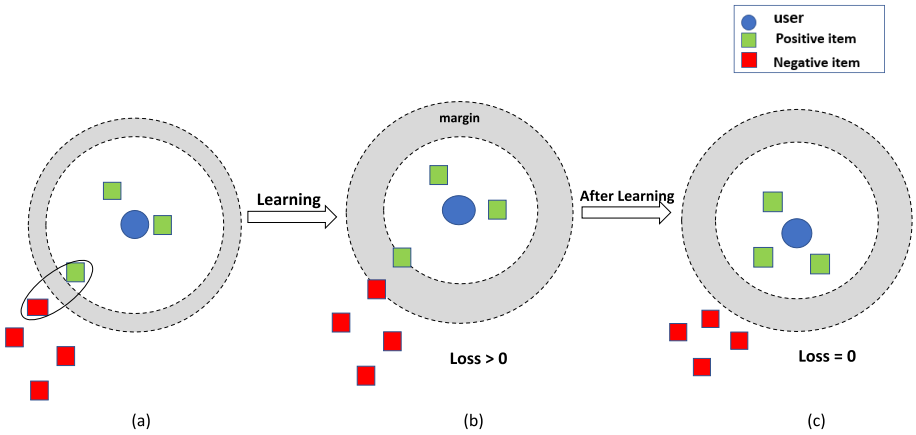


FIGURE 11. Distance learning using hinge distribution loss function. Due to the dynamically evolving margin negative items move inside the margin and the positive item distances including the margin become greater than negative item distances, as shown in (b) . Hence, the hinge distribution loss assigns a positive loss to this pair and improves learning, as in (c).

Similarly, for data preparation, as mentioned in Section IV we created more discriminating and balanced triplets instead of random sampling to make the model more robust against more repeated rating values. In addition, we explored using the random sampling triplet method, which resulted in triplets where most of the items had a rating of 3, because of the imbalance data set problem. By adopting our balanced approach, each item has an equal probability of being part of the triplet, thus, making the model unbiased and transparent toward learning and rating prediction.

B. MODEL LEARNING

As described in Section V, for efficient learning we created a stack formulation of input data. We created this stack formulation by gradually adding contextual information and found that adding more contextual features results in a more information-rich stack and hence improved performance. We also showed that feed-forward neural networks are inefficient at exploring multiplicative relationships between contextual features and hence have little effect on performance. Further, we explored both DNN ablations and CNN ablations in point-wise methods and discovered that DNN with concatenated contextual features did not create much difference in performance improvement, and the embeddings kept the placement of items almost unchanged (see Figure 8a and 8b). In contrast, as shown in Figure 8c and 8d, CNN was able to explore this correlation better than DNN. Moreover, the addition of the proposed grid topology with the pair-wise methods further improved the performance by creating discriminating features (see Figure 8f). As can be observed, the proposed method forms more compact and discriminating clusters by minimizing the distance within a cluster and at the same time maximizing the distances among the clusters. Further, it is worth noting that there were gradual improvements in embeddings from the point-wise to pair-wise method and from the non-grid topology of CNN to the contextual grid topology of CNN. This supports our hypothesis that the proposed method is capable of understanding the preference-based structure of recommendation.

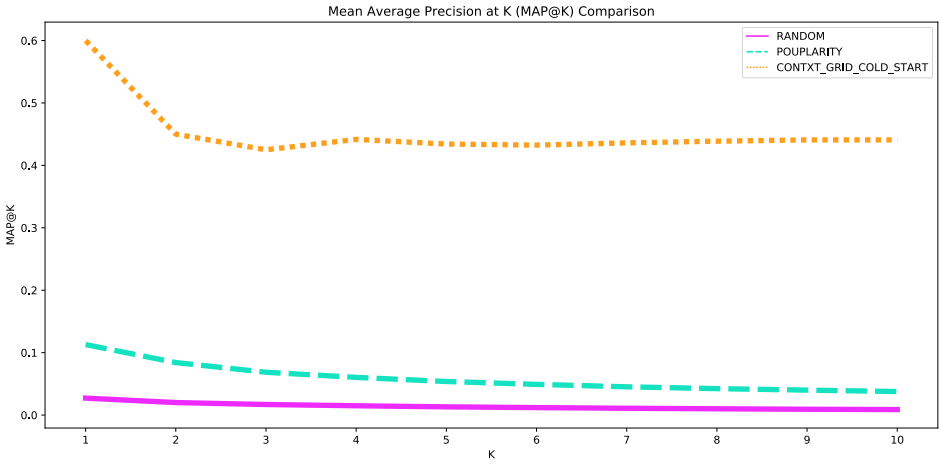


FIGURE 12. MAP@k comparison for random and popularity model with context-driven recommendation model to solve cold start problem.

For learning the embeddings, training was done in batches such that in each batch, a single input data was formulated as $(64 \times 3 \times 150)$-dimensional vector, where 64 is the batch size, 3 is the stack height corresponding to user/item and context $(u/i, c)$, and 150 is stack width corresponding to embedding dimensions. Using kernel sizes 1 and 2 in convolutional layers resulted in the best parameters because the first filter convolves over independent vectors, i.e., user/item or context and then in groups of 2 on $((u/i, c), (c, c))$ vectors, resulting in rich feature representations and thus better modeling performance. Matrix factorization (MF) and other point-wise modeling methods are mainly trained to reconstruct the original matrix, i.e., how well a model is able to reproduce the original ratings, whereas pair-wise methods focus on generating higher probability for more positive items than less positive or negative ones. This preference-based prediction is well-aligned with the recommendation objective. To demonstrate this fact, we chose the top five users from the test set data based on their high count of rated items. We did this in order to show the effectiveness of the proposed method by comparing the embedding space of the proposed method, *Context-Grid-Triplet-Network* and the most relevant ablation variant, *Context-Non-Grid-Triplet-Network*. For visualization, we clustered the embeddings of each user by ratings as shown in Figure 9. As illustrated by the margin lines in Figure 9a, we proved that for a given user, positive and negative items form compact and well-separated sub-clusters from each other by a considerable margin. In conclusion, the model is not only able to differentiate between positive and negative items but also more positive items are well separated from less positive ones, thus, making the model well-suited for an effective and robust recommendation.

C. HINGE DISTRIBUTION LOSS FUNCTION

The existing triplet hinge loss function takes an argument α. This hyperparameter serves as a fixed maximum margin between positive and negative pairs. It remains fixed throughout the training and does not evolve with continuously changing distance among items in embedding space (as a result of model learning), which may lead to inappropriate loss and slow convergence, as illustrated in Figure 10. For example, if some positive items are relatively far away from a user and are closer to the negative items (hard triplets), a hinge loss may not separate the two cases. This would result in a nearly zero loss. Hence, it would be very hard for an algorithm to reduce the distance between a positive item and the user. To solve the aforementioned problem, we propose a hinge distribution loss function, as shown in Equation 15, which incorporates continuously changing (i.e., model learning) positive and negative distance distributions. This helps in creating dynamic and evolving margins for each batch. Thus, each batch improves its margin based upon the latest distance distribution and improves the performance and the convergence, as depicted in Figure 11. We calculate this margin by using Equation 16. First, we compute a batch of positive distances *Bpd* and a batch of negative distances *Bnd* by taking the average of all positive and negative distance distributions of each batch as shown in Equation 14. Secondly, we calculate the margin by taking the difference between two averages and normalizing it as follows:

$$Bpd_b = \frac{1}{2}\sum_{b=1}^{n} L_2(GNet(u, c)) - L_2(GNet(p, c)) \tag{14}$$

$$Bnd_b = \frac{1}{2}\sum_{b=1}^{n} L_2(GNet(u, c)) - L_2(GNet(n, c)) \tag{15}$$

$$\rho_b = \frac{(Bpd_b - Bpd_b)}{(Bpd_b + Bnd_b)} \tag{16}$$

Finally, this margin goes into the loss function, as given in Equation 9. Note that each batch of distances continuously evolves, and as a result of this evolution, positive items will eventually move closer to the user, and their mean distance from the user will gradually decrease. Similarly, negative items will move away from the user, and their mean distance from the user will gradually increase. We use this difference

to create an evolving margin that moves positive items toward the mean of the previous batch of positive items and negative items toward the mean of the previous batch of negative items. By doing this, we also normalize the extreme cases (marginal items). This means that positive items that are relatively closer to the negative ones will be pushed toward the user, and the items already closer to the user will remain inside the margin. Similarly, negative items that are closer to positive ones will be pushed further away, and already distant items will remain outside the margin. This trick also leads to faster convergence, as shown in Table 3 and Table 4. To further optimize this learning process, if the mean of the next *Bpd* is greater than the previous batch, we keep the previous batch mean. The same applies to the *Bnd*.

For learning and performance tracking, we mapped the learned embeddings, from the embedding learner to item labels using both the hinge loss and the hinge distribution loss. Accuracy and AUC are calculated for each model, and we note that the hinge distribution loss improves the accuracy. In addition, we get a quick convergence, which could be achieved even after the first epoch. Also, the margin, ρ, is increased/adjusted gradually for each epoch. The comparison of both loss functions is shown in Table 3 and Table 4. The results are shown only for the first five epochs because the learning behavior stabilizes after that point.

D. SOLVING COLD START PROBLEM

A cold start problem refers to a situation where a model does not have any information about the user. If the user is new to the recommender system and does not have any history of purchased items, then it may be difficult to get recommendations for the user. To avoid this problem we leverage the user's contextual information to predict recommendations for the user. We have jointly trained the user, item, and context embeddings for context-augmented features as described in Section IV. Therefore, at the time of prediction, we only use contextual information to generate ratings for all possible items. Our results in Section 12 show that using contextual information to predict ratings is the better alternative to popularity or random prediction methods. Moreover, grid topology helps to learn item-augmented and user-augmented contextual features, and thus using merely context for prediction reduces the need for user features. We establish the fact that the user's profile as the user's contextual information and learning through grid topology not only improves the recommendation performance but also helps in mitigating cold start problems. It is important to mention here that state-of-the-art methods for solving cold start problems have not been considered in this paper since this is not the scope of this paper. Moreover, we leave this domain to explore in future work.

VIII. CONCLUSION

In this paper, we have highlighted the fact that existing recommendation algorithms fall behind in generating context-driven preferences. This is generally due to the fact that less attention has been paid towards exploring context-augmented features and more attention is paid towards recommendation algorithms itself. To overcome this challenge, we have proposed a Context-Aware approach for top-N recommendations using a *contextual grid triplet network*. This method extends the previously proposed *triplet network* and exploits the CNN architecture to learn context-augmented features (within-grid similarity) and the interactions between users and items (between grid similarity). These two kinds of associations are important for learning low-level and high-level features. Furthermore, Our novel hinge loss variant accelerates convergence by enriching the feature representations within the embedding space and optimizes user-specific preference structure for better recommendations. We have used various state-of-the-art methods that are available to date, and ablation-based experiments to prove the effectiveness and robustness of our proposed approach. Our results have shown that the proposed approach outperforms all baseline methods, including state-of-the-art approaches and ablations variants. Looking ahead, our future work will delve into a detailed exploration of these bi-directional associations using other deep learning architectures, such as Recurrent Neural Networks (RNNs). We remain optimistic about the potential advances this future work could introduce to the realm of context-aware recommendations.

REFERENCES

[1] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit.* Cham, Switzerland: Springer, 2015, pp. 84–92.
[2] F. Aiolli, "Convex AUC optimization for top-N recommendation with implicit feedback," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 293–296.
[3] D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. Dhillon, "Preference completion: Large-scale collaborative ranking from pairwise comparisons," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1907–1916.
[4] H. Ge, J. Caverlee, and H. Lu, "TAPER: A contextual tensor-based approach for personalized expert recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 261–268.
[5] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 79–86.
[6] Z. Gantner, S. Rendle, and L. Schmidt-Thieme, "Factorization models for context-/time-aware movie recommendations," in *Proc. Workshop Context-Aware Movie Recommendation*, Sep. 2010, pp. 14–19.
[7] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, Apr. 2010.
[8] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, and E. H. Chi, "Latent cross: Making use of context in recurrent recommender systems," in *Proc. ACM WSDM*. New York, NY, USA: ACM, 2018, pp. 46–54.
[9] P. G. Campos, F. Díez, and I. Cantador, "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols," *User Model. User-Adapted Interact.*, vol. 24, nos. 1–2, pp. 67–119, Feb. 2014.
[10] Y. S. Rawat and M. S. Kankanhalli, "ConTagNet: Exploiting user context for image tag recommendation," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1102–1106.
[11] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM RecSys*, 2016, pp. 191–198.
[12] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005, pp. 539–546.

[13] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.
[14] R. Meyes, M. Lu, C. Waubert de Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," 2019, *arXiv:1901.08644*.
[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 4th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1998, pp. 43–52.
[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[17] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. NIPS*, 2008, pp. 1257–1264.
[18] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.
[19] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Cham, Switzerland: Springer, 2011, pp. 73–105.
[20] C. Chen, P. Zhao, L. Li, J. Zhou, X. Li, and M. Qiu, "Locally connected deep learning framework for industrial-scale recommender systems," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 769–770.
[21] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1245–1254.
[22] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*.
[23] W. Chen, F. Cai, H. Chen, and M. D. Rijke, "Joint neural collaborative filtering for recommender systems," *ACM Trans. Inf. Syst.*, vol. 37, no. 4, pp. 1–30, Oct. 2019.
[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.* Washington, DC, USA: AUAI Press, 2009, pp. 452–461.
[25] A. Pujahari and D. S. Sisodia, "Pair-wise preference relation based probabilistic matrix factorization for collaborative filtering in recommender system," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105798.
[26] H. Yun, P. Raman, and S. Vishwanathan, "Ranking via robust binary classification," in *Proc. NIPS*, 2014, pp. 2582–2590.
[27] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 193–201.
[28] H. Lu and J. Caverlee, "Exploiting geo-spatial preference for personalized expert recommendation," in *Proc. 9th ACM Conf. Recommender Syst.*, Sep. 2015, pp. 67–74.
[29] J. Manotumruksa, C. Macdonald, and I. Ounis, "A contextual recurrent collaborative filtering framework for modelling sequences of venue checkins," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102092.
[30] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, p. 57, May 2012.
[31] B. Hidasi and D. Tikk, "General factorization framework for context-aware recommendations," *Data Mining Knowl. Discovery*, vol. 30, no. 2, pp. 342–371, Mar. 2016.
[32] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "TFMAP: Optimizing MAP for top-N context-aware recommendation," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2012, pp. 155–164.
[33] R. Misra, M. Wan, and J. McAuley, "Decomposing fit semantics for product size recommendation in metric spaces," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 422–426.
[34] B. Yu, T. Liu, M. Gong, C. Ding, and D. Tao, "Correcting the triplet selection bias for triplet loss," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 71–87.
[35] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2007, pp. 271–278.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[37] F. Vasile, E. Smirnova, and A. Conneau, "Meta-Prod2 Vec: Product embeddings using side-information for recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 225–232.
[38] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[39] A. Livne, M. Unger, B. Shapira, and L. Rokach, "Deep context-aware recommender system utilizing sequential latent context," 2019, *arXiv:1909.03999*.

SOFIA AFTAB received the M.S. degree in IT (data mining) from the National University of Science and Technology, Pakistan. She is currently pursuing the Ph.D. degree in machine learning with the Norwegian University of Science and Technology, Trondheim, Norway. From 2012 to 2018, she was a Data Scientist with different industrial sectors in the domain of recommender systems. Her research interests include information retrieval, deep learning, and text mining.

HERI RAMAMPIARO is currently the Head of the Department and a Professor with the Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Previously, he was the Head of the Data and Artificial Intelligence (DART) Research Group. He has been central in the establishment of the Telenor–NTNU AI-Lab, AI Research Center, NTNU (now Norwegian Open AI-Lab), for which he was a NTNU's Scientific Coordinator. His current main research interests include machine learning, information retrieval, and data/text mining.

HELGE LANGSETH is an expert in the field of computer science with a background as a Statistician, which drives his preference for probabilistic approaches. He is often engaged in probabilistic graphical models, particularly Bayesian networks. His areas of application encompass system monitoring (dynamic models) and the classification of hand-written digits (high-dimensional data analysis). His research interests primarily revolve around machine learning and data mining techniques.

MASSIMILIANO RUOCCO has possesses extensive expertise in machine learning, particularly deep neural networks, and modern artificial intelligence techniques. His focus extends to AI for time series analysis, active learning, and self-supervised learning, with a wealth of skills and experience, he has made significant contributions across various industries, including energy, telecommunications, healthcare, and sports. His work encompasses hardcore machine learning and applied machine learning in real-world contexts.